計算機科学実験及演習3 ハードウェア 「CADツールを用いた設計フロー」

京都大学 工学部情報学科 計算機科学コース 計算機科学実験及演習 3 ハードウェア担当 le3hw@kuis.kyoto-u.ac.jp





- CADツールQuartus Primeを用いた設計フロー
 HDL設計における使い方
 - FPGAへの書き込みと実機上での実行
- ・進め方
 - -加算器を練習として作成してもらいます
 - -HDL設計についてよく知っているのであれば練習は飛ばして、導入課題の7SEG LED駆動回路とカウンタの設計をどんどん進めていって構いません





- •本資料は新演習室環境を想定して作成しています
 - Ubuntu 22.04 LTS
 - Quartus Prime 20.1 Standard Edition
 - ModelSim Intel FPGA Starter Edition
- 各自のPC環境で作業する場合は適宜読み替えてく
 ださい
 - 基本は大きく変わらないはずです
 - 環境違いによる詰まりどころが多いようなら, Slack等で随時お知らせします



FPGAとは?

• Field Programmable Gate Array





FPGAの用途

- ASIC (Application-specific integrated circuit) の設計・検証用LSIに活用
 - ASICの機能を製造前に検証できる
 - -HWの完成前からSWの開発が始められる
- 最終製品に組み込むLSIに活用
 - 民生品での実用例は枚挙に暇がないほど多い
 - 出荷後のデータ書き換え機能を持たせることも
 - 最近の実用例
 - ✓ 金融取引市場, ロボット, データセンター, 機械学習, オート モーティブ(ADAS)



実験環境:FPGAボード

- PowerMedusa MU500-RX/RK
 - FPGA: Altera社Cyclone IV EP4CE30F23I7N (28,840LE)
 - -マイコン:ルネサス社RX210
 - クロック: 20MHz発振器
 - 入出力ユーザインタフェース
 - ✓ プッシュSW x20, ロータリSW x2,
 8bit DIP SW x2, クロックSW x1
 - ✓ 7SEG LED x8, LED x8, ブザー x1
 - 拡張ボード MU500-7SEG

✓ 7SEG LED x64, LED x64





CADツール

- Computer-Aided Design: 論理回路を設計, 合成するツール
 - -環境設定,回路の設計と記述,回路合成, ピンアサイン,シミュレーション, プログラミング(FPGAへの書き込み)までを行う

- EDA(Electronic Design Automation)ツールとも呼ばれる



実験環境:CADツール(実験2と同じ)

- Intel Quartus Prime 20.1を用いる
 - 最新版ではない!
 - Google先生に聞く時にはバージョン番号に注意!
 - ✓ 公式マニュアルやドキュメントも別バージョンはアテにしない
- 各自のPCにもインストール可能
 - 機能制限有りの Lite Edition なら無償利用可能
 - ✓シミュレータはModelSim-Intel FPGA Starter Editionを入れる
 - Starter無しIntel FPGA Editionは有償ライセンスが必要

✓次ページの補足も参照

-Windows/Linuxのみ (macOSは不可)



- 各自PCへのQuartus Primeの導入
 - -機能制限有りの Lite Edition なら無償利用可能です
 - シミュレータ ModelSim-Intel Starter Edition は無償利用可能です
 - ✓ Starter無しの ModelSim-Intel Editionは有償ライセンスが必要
- エディションの違いについて
 - Standard Editionだと30日間限定になります
 - Pro Editionは使用できません (ボード搭載のCyclone VIが未対応)
- 合成性能や最適化結果がエディションで異なってくる可能性があります
 - レポート記載の際には使用した Edition を明記してください



- Quartus Prime 20.1 Lite Editionのインストール方法
 - -Quartus Primeのダウンロードサイト https://www.intel.co.jp/content/www/jp/ja/products/details/fpg a/development-tools/quartus-prime/resource.html
 - ✓ "Legal Disclaimer" で最新版じゃなくて良いの?の警告が 出ることがあるが,無視して "I Agree" すればよい
 - ✓「一式ファイル」の場合は 5.9GB (Linux版 6.4GB)
 - ✓ 「個別ファイル」の場合は下記を同場所にダウンロード

□ Quartus Prime (includes Nios II EDS)

Description ModelSim-Intel FPGA Edition (includes Starter Edition)

Cyclone IV device support

計 3.2GB (Linux版 3.7GB)

-(次ページに続く)



Quartus Primeのダウンロード

- 1. QuartusPrimeのダウンロードページにアクセス。
- 2. Versionから"20.1"を選択
- 3. 「Multiple Download」タブから一式ファイルをダウンロード。または 「Individual Files」タブから個別ファイル(前ページ参照)をダウンロード。

必要なソフトウェアのインストール

- Quartusが動作するためにインストールする必要があるソフトウェアがある。以下にアクセスして、右上の"Download"からPDFドキュメントを取得。
 - ✓ <u>https://www.intel.com/content/www/us/en/docs/programma</u> ble/683472/20-1/introduction.html
- PDFドキュメントの"2.3 Software Requirements"に記載のパッケージをイン ストールする



- Quartus Prime 20.1 Lite Editionのインストール方法
 - 「個別ファイル」の場合は、3ファイルを同位置に配置してください

-インストーラを起動し、適宜チェックを入れて実行してください

- ✓ Installation directory はデフォルトで良いでしょう
 (本資料等での説明はデフォルト位置を想定しています)
- ✓ Select Components では、下記のチェックが必須です

D Quartus Prime (includes Nios II EDS)

Devices > Cyclone IV

D ModelSim - Intel FPGA **Starter** Edition

Windowsの場合は, Quartus本体のインストール終了後に表示される USB Blaster II device driverもインストールしてください

✓ FPGAボードのUSBケーブルがうまく認識しない場合は、
 <u>こちらのページ</u>を参考にしてください.



Quartusの起動

ターミナル上で

\$ /opt/intelFPGA/20.1/quartus/bin/quartus &

- 最後に&を入れるとバックグラウンドプロセスとして実行
 - ✓ https://ja.wikipedia.org/wiki/バックグラウンドプロセス





PATHを通す方法

- ●環境変数(\$PATH)の記述を変更し、コマンドの検索 パスを追加する
- ターミナル上で

\$ export PATH="\$PATH:/opt/intelFPGA/20.1/quartus/bin"

- こうすると、単にquartusと打つだけで起動できるようになる
- ~/.bashrcに上のコマンドを書いておくと、シェル
 (Bash)を起動する度に実行されるので、毎回設定する必要がなくなる





- ライセンス認証(実験2履修者はおそらく不要)
 - 初回起動時, If you have a valid license file, specify the location of your license fileにチェックしてOKを クリック
 - "License file: " に **"27003@is-fpg-10"** を入力するか "LM_LICENSE_FILE variable:" にチェックして入力

- 起動後, Tools -> License Setup..からも設定可能

- Webブラウザの設定
 - Tools -> Options...の Internet Connectivity で, "Web browser:" に好みのブラウザへのパスを設定
 - 何も好みがなければ "/export/share/bin/firefox"

✓ エラー調査時に便利です





- Quartusからmodelsim-aseを呼び出すための 設定をする
 - Tools -> Options... -> EDA Tool Options: ModelSim-Altera: /opt/intelFPGA/20.1/modelsim_ase/linuxaloem
 - ✓ 's' が抜けていることがあるので注意
 - ✓ いったん空白にしてもよい

Options			
Category:			
▼ General	EDA Tool Option	S	
EDA Tool Options Fonts Headers & Footers Settings	Specify the direct	tory that contains the tool executable for each third-party EDA tool:	
	EDA Tool	Directory Containing Tool Executable	
Internet Connectivity	Precision Synt])
Notifications Libraries	Synplify		
▼ IP Settings	Synplify Pro]
IP Catalog Search Location:	Active-HDL])
Design Templates	Riviera-PRO])
Preferred Text Editor	ModelSim])
Processing	QuestaSim]]
Tooltip Settings	ModelSim-Alt	/home/quartus/intelFPGA/17.1/modelsim_ase/linuxaloem]]
Messages Colors	NCSIM		
Fonts	VCS]
	VCS MX])



ツール利用と設計の流れ

- a. 新規プロジェクトの作成と環境設定
- b. HDLコードの作成
- c. コンパイル
- d. タイミング制約の設定と検証 —
- e. シミュレーションによる動作確認
- f. トップレベル回路への入出力ピンの割当て
- g. FPGAへの回路の書き込み

4ビット加算器をお題として設計フローを説明します



練習は組合せ回

路なので無し

練習:4bit加算機の設計



a. 新規プロジェクトの作成と設定(1/4)

- File -> New Project Wizard...
- Introduction "Next"

💿 New Project Wizard

Introduction

The New Project Wizard helps you create a new project and preliminary project settings, including the following:

- Project name and directory
- Name of the top-level design entity
- Project files and libraries
- Target device family and device
- EDA tool settings

You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project.

< Back

Finish

Next >

Cancel

dit <u>V</u> iev	w <u>P</u> roject	<u>A</u> ssignments	P <u>r</u> oces				
		Ctrl+N					
n		Ctrl+O					
		Ctrl+F4					
Project <u>V</u>	<u>/</u> izard						
n P <u>r</u> oject.		Ctrl+J					
e Projec <u>t</u>							
<u>e</u> Project							
		Ctrl+S					
e All		Ctrl+Shift+S					
Properties			-				
ite <u>/</u> Upda	te		•				
ort <u>.</u>							
Convert Programming Files							
e Set <u>u</u> p							
Pre <u>v</u> iew							
		Ctrl+P					
ent F <u>i</u> les			P 💦 IP				
ent Projec	ts		•				
		Alt+F4	Cust				
	dit <u>Viev</u> e Project <u>W</u> Project <u>W</u> Project 2 <u>A</u> All Properties te (Upda ort ₂ Preyiew Preyiew Preyiect Project e Project <u>W</u> Project <u>W</u> All Properties the function of the project Project e Project <u>W</u> e Project <u>W</u> e	dit <u>View Project</u>	idit View Project Assignments Ctrl+N h Ctrl+O e Ctrl+F4 Project Wizard h Project Ctrl+F4 Project e Ctrl+F4 Project e Ctrl+S All Ctrl+Shift+S Properties the / Update ort vert Programming Files e Setup Preview Ctrl+P ent Files ent Projects Alt+F4				





Help

a. 新規プロジェクトの作成と設定(2/4)

• Directory, Name, Top-Level Entity

京都大学

KYOTO UNIVERSITY





a. 新規プロジェクトの作成と設定(3/4)

- Project Type: Empty projectを選択
- Add Files:作成済みの回路をプロジェクトに追加する

Help

- -今回はまだ何も無いので空白のまま"Next"
- Family & Device Settings
 - Family: "Cyclone IV E"
 - Available devices:
 "EP4CE30F23I7"

Available devices:から EP4CE30F23I7を選択

	ect Wizard						
amily & D							
elect the family ou can install a	y and device you want additional device supp	t to target ort with tl	t for com he Install	pilation. Devices comma	nd on the T	ools menu	
Device family			Show in 'Availa	able devices	s' list		
Family: Cyclo	Family: Cyclone IV E		÷	Pac <u>k</u> age:	Any		÷
Devices: All			÷	Pin <u>c</u> ount:	Any		÷
Target device				Sp <u>e</u> ed grade: Any		÷	
 Auto device selected by the Fitter 					er: EP4CE30F		
 <u>A</u>uto device 	e selected by the Fitter	r		Name filter:	EP4CE30F		
 <u>Auto device</u> <u>Specific dev</u> <u>O</u>ther: n/a 	e selected by the Fitter vice selected in 'Availa	r able devic	es' list	Name filter:	EP4CE30F	es 🗌 Ha <u>r</u> o	dCopy compatible only
 <u>A</u>uto device <u>Specific device</u> <u>O</u>ther: n/a <u>v</u>ailable device 	e selected by the Fitter vice selected in 'Availa es:	r able devic	es' list	Name filter: ✓ S <u>h</u> ow adva	EP4CE30F	:s □ Haṟo	dCopy compatible only
<u>Auto device</u> <u>Specific dev</u> <u>O</u> ther: n/a <u>vailable device</u>	e selected by the Fitter vice selected in 'Availa es: Core Voltage	r able devic	es' list	Name filter: Show adva	EP4CE30F Inced device	s 🗌 Haro	dCopy compatible only Idea multiplier 9 bit
 <u>A</u>uto device <u>Specific device</u> <u>O</u>ther: n/a <u>v</u>ailable device <u>Name</u> <u>P4CE30F2317</u> 	e selected by the Fitter vice selected in 'Availa es: Core Voltage	r able devic	es' list	Name filter: ✓ S <u>h</u> ow adva	EP4CE30F	Embed	dCopy compatible only Ided multiplier 9-bit
 <u>A</u>uto device <u>Specific device</u> <u>O</u>ther: n/a <u>vailable device</u> <u>Name</u> <u>EP4CE30F23I7</u> 	e selected by the Fitter vice selected in 'Availa es: Core Voltage 1.2V	r able devic LEc 28848 28949	user 329	Name filter: ✓ S <u>h</u> ow adva	EP4CE30F nced device	Embor 132	dCopy compatible only Ided multiplier 9-bit
 <u>A</u>uto device <u>S</u>pecific dev <u>O</u>ther: n/a <u>v</u>ailable device <u>Name</u> <u>P4CE30F23I7</u> <u>Companion device</u> 	e selected by the Fitter vice selected in 'Availa es: Core Voltage 1.2V	r able devic LEc 28848 28949	User 329	Name filter: ▼ S <u>h</u> ow adva 608256 608256	EP4CE30F nced device	Embed	dCopy compatible only Idea multiplier 9 bit
<u>Auto device</u> <u>Specific device</u> <u>O Other:</u> n/a <u>vailable device</u> <u>Name</u> <u>EP4CE30F23I7</u> <u>Companion device</u> HardCopy:	e selected by the Fitter vice selected in 'Availa 25: Core Voltage 1.2V Lov Vice	r bble devic LEc 28848 20040	uses' list	Name filter: ▼ S <u>h</u> ow adva 5. 008256 608256	EP4CE30F nced device	Embod	dCopy compatible only Ided multiplier 9-bit

< Back

Next

選択が面倒な場合は フィルタ検索も可能



Cancel

Finish

a. 新規プロジェクトの作成と設定(4/4)

New Project Wizard

- EDA Tool Settings
 - -外部の設計ツールを使用/ 変更する場合に指定する
 - "Simulation"にて使用する HDLを指定する
- Summary
 - 設定を確認して"Finish"

New Project Wizard Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings

Project directory:	/home/lab3/takase/le3a/adder4bit		
Project name:	adder4bit		
Top-level design entity:	adder4bit		
Number of files added:	0		
Number of user libraries added:	0		
Device assignments:			
Family name:	Cyclone IV E		
Device:	EP4CE30F23I7		
EDA tools:			
Design entry/synthesis:	<none> (<none>)</none></none>		
Simulation:	ModelSim-Altera (Verilog HDL)		
Timing analysis:	0		
Operating conditions:			
VCCINT voltage:	1.2V		
Junction temperature range:	-40-100 °C		

< <u>B</u>ack

Next

Cancel

Finish

EDA LOOIS:					
Tool Type	Tool Name		Format(s)		Run Tool Automatically
Design Entry/S	<none></none>	1	<none></none>	1	Run this tool automatically to synthesize the current
Simulation	ModelSim-Altera	\$	Verilog HDL	_\÷	Rt n gate-level simulation automatically after compared and a simulation automatically after a simulation automatically after and a simulation automatically after and a simulation automatically after and a simulation automatically after a simulation automatically after a simulation automatically after and a simulation automatically after a simulation automatically after a simulatically after a simulation automatically after a sim
Formal Verific	<none></none>	Ŧ		Le .	
Board-Level	Timing		<none></none>	\$	
	Symbol		<none></none>	\$	
	Signal Integrity		<none></none>	\$	
	Boundary Scan		<none></none>	\$	





- ブロック図によって加算器を設計した
 –これだけでも手間がかかった
- 実験3ではハードウェア記述言語を用いることで飛躍的に生産性が高まる





b. HDLコードの作成(1/3)

本実験はVerilog-HDL前提

他の言語だと教員・TAもわからない可能性大

- HDLエディタを使用する -File -> New -> Verilog HDL File または File -> New -> VHDL File ✓ 自分の使いたいHDLに合わせて選択する - 実験3ではハードウェア記述言語による設計を 存分に楽しんでください ✓ 回路図との組合せもOKです ✓ トップレベルは回路図,部品はHDLとすると 全体の見通しが良くなります • まずは(泣く前に)HDLファイルを保存 -File -> Save As... 数字から始まる ファイル名は避ける -"adder4.v"
 - ✓ VHDLの場合は "adder4.vhd"

京都大学

KYOTO UNIVERSITY

• Tasks窓のFlow:を "Compilation"にする

😣 🗊 New
New Quartus II Project
Design Files
AHDL File
Block Diagram/Schematic File
EDIF File
Qsys System File
State Machine File
SystemVerilog HDL File
Tcl Script File
VHDL File
Verilog HDL File
Memory Files M
Hexadecimal (Intel-Format)
Memory Initialization File
- Verification/Debugging Files
In-System Sources and Prob
Logic Analyzer Interface File
SignalTap II Logic Analyzer F
University Program VWF
Other Files
AHDL Include File
Block Symbol File
Chain Description File
Synopsys Design Constraint
- Text File
Cancel Help

Save A	s mighta ball the set of the set	• 0 0	0 🖗	
Compute	r db			
File <u>n</u> ame:	adder4.v			Save
Save as type:	Verilog HDL Files (*.v *.vlg *.verilog) ✓ Add file to current project		\$	Cancel



b. HDLコードの作成(2/3)

- (初期)ウインドウの構成
 - 使いやすいようにレイアウトやサイズ等を適宜調整してよい View -> Utility Windows など
 - エディタ機能は貧弱なので,ぜひ好みのエディタを使用しよう (外部ツールとのファイル同期は自己責任で)



b. HDLコードの作成(3/3)

• Verilog HDLの例

	adder4.v 🗶
	4 🖉 0 🔊 🖉 🖓 0 👘 4 🖉 1 🕸
1 2 3 4 5 6 7 8	<pre>module adder4 (input [3:0] a, b, input cin, output [3:0] sum, output cout); assign {cout, sum} = a + b + cin; endmodule</pre>

• VHDLの例

•	adder4.vhd 🗙	
	# 🕼 📅 👍 👘 🛍 🖉 👘 🖓 #	2
1	library IEEE;	
2	<pre>use IEEE.std_logic_1164.all;</pre>	
3	<pre>use IEEE.std_logic_unsigned.all;</pre>	
4		
5	entity adder4 is	
6	port (
7	<pre>a, b : in std_logic_vector(3 downto 0);</pre>	
8	cin : in std_logic;	
9	<pre>sum : out std_logic_vector(3 downto 0);</pre>	
10	<pre>- cout : out std_logic);</pre>	
11	end adder4;	
12	L	
13	<pre>architecture adder4_body of adder4 is</pre>	
14	<pre>signal tsum : std_logic_vector(4 downto 0);</pre>	
15	Degin	
16	tsum <= ('0' & a) + ('0' & b) + cin;	
17		
18	cout <= tsum(4);	
19	<pre>sum <= tsum(3 downto 0);</pre>	
20	end adder4_body;	
21		

2



c. コンパイル (1/4)

- FPGA内の論理ブロックの構成情報に変換する
 - 回路図をレジスタ転送レベルに変換する
 - 論理合成, テクノロジマッピング, 配置配線を行う
 - ✓ RTLで記述された回路をFPGA上の論理ブロックに割り付け, 論理ブロックの配置や配線を行う
 - ビットストリームを生成する

✓ 配置/ 配線の情報をバイナリ記述の構成情報ファイルに変換する

- コンパイル手順
 - Processing -> Start Compilation または ツール操作アイコンのStart Compilation または Ctrl+L

✓ Start Analysis & Synthesis で入力チェックのみも可能



c. コンパイル (2/4)

- Tasks窓の"Compilation"がall green v になったら正常にコンパイル成功
 - 黄字?はそのフローで再コンパイルが必要なことを表す
 - -赤字×はそのフローでエラーが発生していることを表す
- 正常終了しなかったら、下部のメッセージウインドウまたは "Compilation Report -> Flow Messages" などを確認し, メッセージをもとにデバッグを行う
 - 右クリック Help でメッセージのWeb検索可能
 - Critical Waning / Warning も要チェック
 - ✓ CADツールでは警告がシビアに出力されがちだが,無視する場合も警告の 意味を理解しておくこと(設計ミスに気づくことも多い)

(() •)
	All S A V < S S S S S S S S S S S S S S S S S S S	
T	Type ID Message	-
	0 204019 Generated file adder4.vo in folder "/export/home/lab3/le3a/TA/yano/le3atest/simulation/modelsim/" for EDA simulation tool	
	• 204019 Generated file adder4_7_1200mv_100c_v_slow.sdo in folder "/export/home/lab3/le3a/TA/yano/le3atest/simulation/modelsim/" for EDA simulation tool	
	• 204019 Generated file adder4_7_1200mv40c_v_slow.sdo in folder "/export/home/lab3/le3a/TA/yano/le3atest/simulation/modelsim/" for EDA simulation tool	
	0 204019 Generated file adder4_min_1200mv40c_v_fast.sdo in folder "/export/home/lab3/le3a/TA/yano/le3atest/simulation/modelsim/" for EDA simulation tool	
	0 204019 Generated file adder4_v.sdo in folder "/export/home/lab3/le3a/TA/yano/le3atest/simulation/modelsim/" for EDA simulation tool	
Þ	🛛 🛛 Quartus Prime EDA Netlist Writer was successful. 0 errors, 1 warning	_
es	0 293000 Quartus Prime Full Compilation was successful. 0 errors, 13 warnings	=
Sag		
les	System Processing (127)	
2 ~	······································	



28

c. コンパイル (3/4)

- Compilation Report で合成結果を確認する
 - -Processing -> Compilation Report または Ctrl+R
 - 各項目の詳細は各自で調査すること

🔶 adde	r4.v 🗙	Compilation Report - adder4	4bit 🗙
Table of Contents	Р (б	Flow Summary	
Flow Summary Flow Settings Flow Non-Defau Flow Elapsed T Flow CS Summ Flow OS Summ Flow Log Flow Log Flow Log Flow Log Flow Log Flow Log Flow Log Flow Log Flow Sypresse Flow Suppresse	ult Global Settings ime ary thesis ing Analyzer iter ed Messages	Flow Status Quartus II 32-bit Version Revision Name Top-level Entity Name Family Device Timing Models Total logic elements Total combinational functions Dedicated logic registers Total registers Total pins Total virtual pins Total wirtual pins Total memory bits Embedded Multiplier 9-bit eleme Total PLLs	Successful - Wed Apr 12 16:20:56 2017 13.0.1 Build 232 06/12/2013 SP 1 SJ Full Ver: adder4bit adder4 Cyclone IV E EP4CE30F23I7 Final 6 / 28,848 (< 1 %) 0 / 28,848 (< 1 %) 0 14 / 329 (4 %) 0 0 / 608,256 (0 %) ents 0 / 132 (0 %) 0 / 4 (0 %)
			3



c. コンパイル (4/4)

- 回路サイズ: Fitter -> Summary
 - FPGA上の各ブロックの使用量
 - -100%を超える項目があるとFPGAデバイスに収まらない
- 動作速度と遅延: Timing Analyzer
 - SDCでset_max_delay, set_min_delayを設定すればそれぞれ の入力端子から出力端子への伝搬遅延時間(単位はns)を表示で きる(FAQ参照)
 - 順序回路の設計で入力クロックを指定すれば,動作周波数なども 表示できるようになる
 - 赤字の項目がある場合はタイミング制約を満たしていない
 - ✓ Unconstrained Pathsは set_max_delay, set_min_delayを 適切に設定すれば消える 順序回路設計では特に気にする必要がある



d. タイミング制約の設定(参考)

- ・練習の加算器は組み合わせ回路なので設定の必要はないが、クロックを使用する順序回路では必ず設定する。
- クロックの指定やタイミング制約(レジスタ入力へのセットアップ 時間やホールド時間, 伝搬遅延など)の設定を行う
 - -基本的なクロックの指定のみ解説,詳細は各自で調査すること.
- Timing Analyzerを使用する
 - Tools -> Timing Analyzer または, ツール操作アイコンの "Timing Analyzer"

	🥪 🗇 🕤 TimeQuest Timing Analyzer - /home/lab3/takase/le2hw3/training/dfftest/dfftest - dfftest
	Eile View Netlist Constraints Reports Script Tools Window Help 🕏
	Report 4 🛛 Getting Started 0 🕂
	Report not available Welcome to the TimeQuest Timing Analyzer
	Tasks
泉都大字	
す、 トレ ノ チーオ	- Type "help" to view a list of Quartus II Tcl packages.
(YOTO UNIVERSITY	- ype near yperage name to the size of the command available for the specified Quartus II Tcl package. Tome "helm -tcl" to get an overview on Quartus II Tcl package.

d. タイミング制約の設定(参考)

- Timing Analyzer
 - Tasks窓の "Netlist Setup -> Create Timing Netlist" を実行

e Path

- Tasks窓の "Netlist Setup -> Read SDC File" を実行
- Constraints -> Create Clock... からクロックを設定

	11-								
Eile View Netlist Constraints Br	- /nc	me/lar			🔲 Tim	eQuest '	Timing Analy	zer - /hom	e/lat
	ep <u>o</u> n	s <u>s</u> u		<u>F</u> ile	<u>V</u> iew	<u>N</u> etlist	<u>C</u> onstraints	Rep <u>o</u> rts	<u>S</u> cr
Report 40		SDC	(Repor	t		Create Cl	ock	
SDC File List	1	dfftes	5		Time	Quest Tir File List	Create Ge Set Clock Set Clock	enerated C Latency Uncertaint	юск. ty
							Set Clock Remove (Groups Clock	
Tasks #0 R Company Project				•			Set Input Set Outpu	Delay ıt Delay	
Reset Design				Tasks	Open F Netlist S	Project Setup	Derive PL Derive Cle	L Clocks ock Uncert	ainty
Set Operating Conditions Peports Carlow Slack The Construction of the Construction Report Hold Summary The Report Recovery Summary The Report Removal Summary The Report Minimum Pulse Wid The Report Max Skew Summar					Creating Read Reset D Set Ope Reports Set Ope Reports Slack	ate Timin d SDC Fil ate Timin Design erating Co k	Set False Set Multic Set Maxin Set Minim Set Max S	Path :ycle Path num Delay num Delay. ikew	
Datasheet Totasheet Totasheet Totasheet Totasheet Totasheet Device Specific						eport Sett eport Holo eport Rec	Read SDC Write SDC	File File	
Report TCCS					- 📆 R	eport Min	Reset Des	sign	

京都大学 KYOTO UNIVERSIT

d. タイミング制約の設定(参考)

- Timing Analyzer
 - Constraints -> Create Clock... からクロックを設定
 - ✓ Clock name: クロック名の定義 未記入でも可(例: clock)
 - ✓ Period: クロック周期の設定
 - ✓ Waveform edges: 立ち上がり/立ち下がり時間の設定
 - ✓ Targets: 設計回路上のクロックを選択
 - "List"から表示して">"で選択
 - ✓ SDC commandに生成される 制約が表示される

	No duplicates
😣 🗉 Create Clock	Matchas
Clock name: Period: 10.000 ns	List 14 matches
Waveform edges Rising: ns	Q1[3] • > lock Q2[0] Q2[1] >> > Q2[2] >> >> >
Falling: ns 0.00 5.00 10.00 Targets:	Q2[5] Clock dsta[0] data[1] data[2]
SDC command: create_clock -period 10.000	SDC command: [get_ports {clock}]
Run Cancel Help	OK Cancel I

Name Finder

Collection: get ports

Case-insensitive

Hierarchical
Compatibility mode

Options

♦ Filter: *



elp

d. タイミング制約の設定(参考)

- Timing Analyzer
 - Reports/ 以下でタイミングに関する種々のレポートが 報告されるようになる
 - ✓ 満足していないものがあれば制約の追加などで対処する
 - ✓ Compilation Report からも確認できるようになる
 - ✓ Fmax Summary:動作可能な最大周波数(与える制約によって変わる)
 - ✓ Clocks: 現在設定値での設計回路のクロック周波数
 - Tasks窓の "Write SDC File…" で制約ファイルを保存
 - ✓ <module>.sdc とするとプロジェクトに自動認識される (.outを取ったほうがよい)
 ◎ Write SDC File
 - ✓ 慣れてきたらテキストベースで 制約ファイルを作成して編集してもよい

😣 🗊 Write SDC File
SDC file name: dfftest.sdc
✓ Expand
Tcl command: write_sdc -expand "dfftest.sdc"
OK Cancel Help



e. シミュレーション(1/11)

- まずはテストベンチファイルを作成する
 - テストベンチファイルとは,設計した回路が期待通りの動作をする か確認するためのファイルのこと.
 - ハードウェア記述言語Verilog HDLで記述する(実験3では文法の理解も すること)
- 回路に任意の入力を与え、シミュレータModelSimで出力を観測する





e. シミュレーション (2/11)

- テストベンチファイルを作成する手順
 - Processing -> Start -> Start Test Bench Template Writer
 - ./simulation/modelsim/*.vt が作成される
- *.vt を編集 もしくは コピーして新しいファイルを作る
 - ✓ .vにするとエディタによってはverilogモードが使える
 - ✓ 時間単位を変更する
 - `timescale 1 ns / 1ps
 - □ 1タイムステップが1ns、丸め精度(最小単位)が1psという意味
 - ✓ 以下の記述の直下に初期設定を記述する
 - // code that executes only once
 - // insert code here --> begin
 - ✓ 以下の記述の直下に所望の値(波形)を記述する
 - // code that executes for every event on sensitivity list
 - // insert code here --> begin
 - □ eachvec信号は使わないので@eachvec;はコメントアウトか削除する

慣れてきたらTemplate Writerを使わなくても良い



e. シミュレーション (3/11)

テストベンチファイルの記法例
 -代入:

 A <= 0;
 B <= B + 1;
 Cin <= ~Cin; //反転
 As <= 4'b10_01; // 4ビッ
 トの2進数 (binary)
 ✓ (遅延が無ければ)並列に
 代入される

Verilog-HDLで書かれています。 何も指定しないと数字は十進数となります。

京都大学

KYOTO UNIVERSITY

-**遅延:** #1000

- **繰り返し:** always begin #100

end

✓ クロック波形の生成に便利

- ✓ initial begin end や always begin - end の外側, module - endmodule の 内側に記入すること
- ✓ alwaysブロック同士は
 並列に実行される
- ✓ eachvecは未使用のため @eachvec;があると繰り返 されない

e. シミュレーション (4/11)

- テストベンチファイルの記述例
 時間単位も変更済み
 - `timescale 1 ns / 1ps
 - -単位や遅延はよく考える ✓ あまりに小さいと??

```
initial
begin
// code that executes only once
// insert code here --> begin
a <= 4'b0000;
b <= 4'b0000;</pre>
cin <= 0;
// --> end
$display("Running testbench");
end
always
// optional sensitivity list
// @(event1 or event2 or .... eventn)
begin
// code executes for every event on sensitivity list
// insert code here --> begin
#100
a <= 4'b1100;
b <= 4'b0010;
cin <= 0;
#100
a <= 4'b0110;
b <= 4'b1010;
cin <= 0;
#100
a <= 4'b0110;
b <= 4'b0011;</pre>
cin <= 1;</pre>
#100
a <= 4'b1001;
b <= 4'b0100;
cin <= 1;
//@eachvec;
```

// --> end
end
endmodule

);

@eachvec;は コメントアウト



e. シミュレーション (5/11)

- Quartusからmodelsim-aseを呼び出すための設定をする
 - -Assignment -> Settings -> EDA Tool Settings -> Simulation
 - ✓ Tool name: ModelSim-Altera
 - ✓ Format for output netlist:
 - Verilog HDL (VHDLでの設計時はVHDL)
 - ✓ NativeLink settings: Compile test bench
 - ✓ Test Benches... をクリック

🔵 🕕 Settings - fulladder	
ategory:	Device
General	Simulation
General Files Libraries Operating Settings and Conditi Voltage Temperature Compilation Process Settings Early Timing Estimate Incremental Compilation Physical Synthesis Optimizar EDA Tool Settings Formal Verification Board-Level Analysis & Synthesis Settings VHDL Input Verlig HDL Input Default Parameters Fitter Settings Fitter Settings Assistant	Simulation Specify options for generating output files for use with other EDA tools. Tool name: ModelSim-Altera Run gate-level simulation automatically after compilation EDA Netlist Writer settings Format for output n tilst: Verilog HDL Time scale: 1 ps Output girectory: simulation/modelsim Map illegal HDL characters Enable glitch filtering Options for Power Estimation Generate Value Change Dump (VCD) file script Settings Design instance name: More EDA Netlist Writer Settings
SignalTap II Logic Analyzer	NativeLink settings
– Doger Analyzer Miteriace – PowerPlay Power Analyzer Sett – SSN Analyzer	None Compile test bench: Use script to set up simulation:
	O Script to compile test bench: More NativeLink Settings Reset OK Cancel Apply Help



e. シミュレーション (6/11)

– Test Benches



- ✓ New... をクリック
 New Test Bench Settings
 - ✓ Test bench name: 任意の名前
 - ✓ Top level module in test bench: を 実際のテストベンチモジュール名にする (Template writerを使った場合, *_vlg_tst になっている)
 - ✓ Test bench and simulation files: simulation/modelsim/*.vt
 - (…から前ページで作成したファイルを選択)
 - ✓ Add をクリック
- Test Benchesに戻ってOK, Applyをクリック

🙁 🗉 Edit Test Bench Settings	
Edit test bench settings for the selected test bench.	
Test bench name: test1	
Top level module in test bench: test1	
$\hfill\square$ Use test bench to perform VHDL timing simulation	
Design instance name in test bench: NA	
Simulation period	
$\ensuremath{}$ Run simulation until all $\underline{v}ector$ stimuli are used	
○ <u>E</u> nd simulation at:	
Test bench and simulation files	
Eile name: simulation/modelsim/adder4_test1.vt	<u>A</u> pp V
File Name Library HDL Version	<u>R</u> emove
	<u>U</u> p
	Down
	Properties
OK Cancel	Help





e. シミュレーション (7/11)

- 設計を再コンパイルする(タスク窓が?になっているはず)
- シミュレータを起動する

-Tools -> Run Simulation Tool -> Gate Level Simulation...

w3/	training/fulladder/fulladder - fulladder		
ng	<u>T</u> ools <u>W</u> indow <u>H</u> elp 🐬		
	Run Sim <u>u</u> lation Tool	RTL Simulation	
	え Launch Simulation Library <u>C</u> ompiler	🦎 <u>G</u> ate Lev <mark>y</mark> l Simula	tion
.08	🗞 Launch Design Space E <u>x</u> plorer	Summany	
	📋 TimeQuest Timing Analyzer	Status	Successf
	Advisors	rtus II 32-bit Version sion Name	13.0.1 Bu fulladder
	 Chip Planner Design Partition Planner Netlist Viewers 	level Entity Name ily ce	fulladder Cyclone EP1C6Q2 Einal
	SignalTap II Logic Analyzer In-System Memory Content Editor Logic Analyzer Interface Editor In_System Sources and Probes Editor SignalProbe Pins	l logic elements l pins l virtual pins l memory bits l PLLs	2 / 5,980 5 / 185 (0 0 / 92,16 0 / 2 (0 %



e. シミュレーション (8/11)





e. シミュレーション (9/11)

- 波形が表示されない場合
 - テストベンチファイルを念のため再コンパイルしよう
 - ✓ "Library" ウインドウの gate_work -> <module>_vlg_tst を 右クリックして "Recompile"
 - Transcriptionウィンドウに# Error loading design"が出 ているか確認
 - ✓ 再コンパイルしても出ている場合、テストベンチファイルにミスがあるか、ファイルの指定が間違っているなどの可能性がある

😣 🖨 💷 Mod	elSim ALTERA STARTER EDITI	ON 10.1d				
<u></u> Eile <u>E</u> dit ⊻iew	<u>C</u> ompile <u>S</u> imulate A <u>d</u> d Library T	<u>o</u> ols Layo <u>u</u> t Boo <u>k</u> marks <u>W</u> indow				<u>H</u> elp
📄 - 🚅 🔛 -	s 🚳 🐰 🖻 🛍 🖄 🖄 🚳 s	A 🗄 🖬 🕼 🏶 🗰 🛺 🐹 👘			* * * * * *	
Layout Simula	ce 🗾 🖌 ColumnLayout 🏾	llColumns	- 🛛 🧠 🖧 - 🥵 🗍 🗖	I O 10 🖸 📶 🥢 🕅 🕅		J L I F J
] 3≪ + ⇒€ + ∰•	Search:	1 R B G G G G G G		/ <i>/</i>		
🔥 Library 🚃	💳 🕂 🗗 🗙 👔 Objects 💷 🔤		Default			+ # ×
▼ Name	Name 🛛	Value Kind 💊 🗸	Msgs			
	VIg_ Simulate omorace without Optimization Simulate with full Optimization Simulate with Coverage Edit Ver Refresh					
京都大学 KYOTO UNIVERSII	Create Wave					

e. シミュレーション(10/11)

- 波形が表示されない場合(その2)
 - -Simulate -> Start Simulation...
 - Design タブの "Design Unit(s)" にて gate_work.<module>_vlg_tst を選択
 - Libraries タブの "Search Libraries" にて Add... して 必要なライブラリを追加する ("cycloneive_ver"のみ選択でよい)
 - ✓ "altera_ver", "gate", "gate_work" が自動的に入っていることが あるが, どちらでもよい

😣 🗆 🗈 🛛 Model Sim A	LTERA S	TART	ER EDI	TION 1	10.1d
$\underline{F}{ile} \underline{E}{dit} \underline{V}{iew} \underline{C}{ompile}$	<u>S</u> imulate	A <u>d</u> d	Lįbrary	T <u>o</u> ols	Layo
🛛 🖻 • 🗃 🖬 🛸 🚳 🏲	Design O Start Simi	ntimiza Ilation	tion	-	⊨м
Layout Simulate	Runtime	ptions	 3	.Co	lumns
] 3•• • •€ • ∰• Search	<u>R</u> un			► 100	×
🏦 Library 🚃 🋲 🖬	Step			► we	
▼ Name	Restart				Va
●_#L work ■_#L gate_work ● fulladder	<u>B</u> reak End Simu	lation	B	Ireak	

京都大学

KYOTO UNIVERSITY



44

e. シミュレーション(11/11)

- その他の操作
 - "Objects" ウインドウから観測したい信号線を "Wave" ウインドウにドラッグする
 - "Run Length" を所望の大きさにする(例:100ns)
 ✓ 大きすぎるとシミュレーション時間が掛かる
 - "Run" や "Run All" などをクリックする
 - ✓ "Transcript" ウインドウで直接入力も可能 (慣れてきたらこっちのほうが早いかも)
 - 波形の表示は右クリックまたはキー入力で Zoom In/Out/Fullなどが変更可能
 - -波形はテストベンチで指定済みだが,直接指定も可能

✓ 信号線の右クリックで"Force..."など

- テストベンチを変更した場合は, "Library" ウインドウからRecompileして "Restart"



f. 入出カピンの割当て(1/3)

- 回路の信号線をFPGAの入出カピンに割り当てる
 - -割当ての指定に応じて、ボード上のスイッチから入力を与えたり、 LEDに信号を出力したりできるようになる
 - どのモジュールがどの入出力ピンに対応しているかは ユーザーズマニュアルとピンアサイン表 を参照すること
 - ✔ ピンアサイン表まとめ が非常に便利
 - https://panda.ecs.kyoto-u.ac.jp/access/content/group/2024-110-9084-000/HWマニュアル/ピンアサイン表_rev.xlsx
- FPGAの入出カピンと各種装置との対応は, 取扱説明書およびピンアサイン表を精読する
 - 今回の例では、2つの入力値をディップSWのA,B,
 キャリーインをテンキーのプッシュSW SW4,
 和とキャリーアウトをLEDとしている

- 上記ピン割当ては使い勝手に応じて自由に変えてみましょう



f. 入出カピンの割当て(2/3)

- Pin Plannerを起動する
 - -"Assignments" -> "Pin Planner" または Ctrl+Shift+N





f. 入出カピンの割当て(3/3)

- 入出カピンを割り当てていく
 - Node NameとLocationを対応づける
 - ピンの名前が出てこない場合は設計を再コンパイルする
 - -別のピン名が出る場合は、Top-levelになっているか確認する



f. 入出カピンの割当て(3/3)

ディップスイッチ

- 今回の割当て例
 - ディップスイッチで入力を与える
 - 結果をLEDで表示させる
- ・ピンアサイン表まとめを見ながら
 やってみよう
 - https://panda.ecs.kyotou.ac.jp/access/content/group/2024 -110-9084-000/HWマニュアル/ピン アサイン表_rev.xlsx



LEC



- 合成した回路イメージをFPGAに書き込み,実機上で動作を検証する
 - 実機で動作して初めて設計の完了!
 - Tasks窓が黄字?の場合は再コンパイル, 赤字×の場合はエラーなのでHDLコード等をデバッグする



- FPGA書き込みツールを起動する
 - "Tools" -> "Programmer" または Tasks窓の"Program Device (Open Programmer)"

😣 🗐 💿 Quartus II 32-bit - /home/lab3/takase/le3a/ad	ler4bit/adder4bit - adder4bit		
File Edit View Project Assignments Processing Image: Image	Tools Window Help Image: Plan Simulation Tool Run Simulation Tool Launch Simulation Library Compiler Launch Design Space Explorer	Compilation Report - adder4bit	Search altera.com
[™] adder4.v	<u>T</u> imeQuest Timing Analyzer <u>A</u> dvisors <u>A</u> dvisors <u>O</u> Chip Planner <u>D</u> Design Partition Planner Netlist <u>V</u> iewers	Flow Status Quartus II 32-bit Version Revision Name Top-level Entity Name Family Device Timing Models	Successful - Wed Apr 12 17:51:01 2017 13.0.1 Build 232 06/12/2013 SP 1 SJ Full Vers adder4bit adder4 Cyclone IV E EP4CE30F23I7 Final
	 SignalTap II Logic Analyzer In-System Memory Content Editor Logic Analyzer Interface Editor In-System Sources and Probes Editor SignalBrobe Pins Programmer 	Total logic elements - Total combinational functions - Dedicated logic registers Total registers Total pins Total virtual pins Total memory bits Ernbedded Multiplier 9-bit elements Total virtual pins	6 / 28,848 (< 1 %) 6 / 28,848 (< 1 %) 0 / 28,848 (0 %) 0 14 / 329 (4 %) 0 0 / 608,256 (0 %) 0 / 132 (0 %)
→ Hierarchy Files J Design Units 1 P Co	2 TAG Chain Debugger Transceiver Tool <u>k</u> it		0/4(0%)
Flow: Compilation ↓ Customize. Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Compile Design Image: Com	External Memory Interface Toolkit MegaWizard Plug-In Manager Nios II Software Build Tools for Eclipse Qsys Tcl Scripts Customize Options License Setup Install Devices		



- FPGAボードをPCに接続
 - 電源に接続
 - -USBを接続



- Programmerの設定を行う
 - -Hardware Setupをクリックして, Currently selected hardware: を "USB-Blaster [X-X]" とする
 - ✓ Available hardware itemsにUSB-Blasterが出てこない時は, PCとUSB接続し直す. それでも出てこない場合はデバドラの 問題も考えられるのでTAまで.
 - -Modeを"JTAG"にする
 - -リストにファイルが表示されて いない場合は, Add File... から 書き込むファイルを選択する
 - <proj_name>.sof
 - リスト上のProgram/Configureに チェックを入れる



数字は環境に

よって異なる







- FPGAに回路を書き込む
 - Startをクリック
 - Process: が緑の"100% (Success)"になったら書き込み完了
 - FPGAボードが認識されているのに、書き込みがうまくいかない(Failed になる)場合,以下のコマンドを実行(jtagdを再起動)してもう一度試す
 - ✓ killall -9 jtagd
 - ✓ /opt/intelFPGA/20.1/quartus/bin/jtagconfig
 - □ エラーが出ないことを確認
 - ✓ 詳細は https://isle3hw.kuis.kyoto-u.ac.jp/faq.html#programmer_failed を参照
- 実機上で動作を検証する
 - 正しく所望の動作ができて いれば完了!
 - ディップスイッチは負論理なので注意
 - ✓ ONで0になる



😣 🖻 🗈 🛛 Progra	mmer - /home/lab3/takase	/le3a/adder4bit/	/adder4bit - a	adder4bit -	[output_fil	es/adde	er4bit.co	
<u>File Edit Vie</u>	w P <u>r</u> ocessing <u>T</u> ools <u>W</u> i	indow <u>H</u> elp ⁽⁾	Ð		Searc	:h altera	.com	٩
Hardware Se	etup USB-Blaster [3-1] me ISP to allow background p	Mode: JTA	G MAX II and M	1AX V devic	Progr es)	ess: 0%	6 (Succe	5
M Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Ex
	output_files/adder4bit.sof	EP4CE30F23	001FC551	001FC551	₹			
Auto Detect								
🗙 Delete								
Add File	4	III						Þ
Change File.								
Save File		٦						
- Add Device								

- うまくいかない場合はまずFAQを確認。
 - 過去にあった様々な質問とそれに対する回答がまとめられています。
- 実機で思った通りに動かない
 - まずはシミュレーションで論理的に正しく設計できているか, きちんと確認しましょう
 - シミュレーションが上手くいっていても実機で動かないことも よくあります. 多くはタイミング制約を満足できていないことに 起因します. FAQを参考にしてください.
- 入出力装置が思った通りに動かない
 - もちろん設計が正しいかは当然として,入出力装置には それぞれ正論理と負論理があります
 - 7SEG LEDはセレクタ信号の設定も必要です
 - 未割り当てのLED(ピン)が光るのは無視して構いません
 - ✓ きちんと考慮するのが綺麗ですが,,,,
 - 詳しくは要するに取扱説明書を精読してください

- 実機上で色々な情報を一度に出力したい
 - 拡張ボードMU500-7SEGを使いましょう

✓ 使い方の詳細は取扱説明書を参照してください

- 拡張ボードを使いこなせば、プロセッサのPC現在値だったりを 実機上で確認できるようになってデバッグが非常に捗ります
- HDLの回路を回路図エディタから呼び出したい
 - 対象ファイルを選択した上で "Files" -> "Create / Update" -> "Create Symbol Files for Current File"を選択する
 - ✓ 回路図エディタから部品インスタンスとして配置できる

✓ 逆もできます(回路図をHDLから下位モジュールとして呼び出し)

- -プロセッサ設計では,全体のアーキテクチャとデータパスは 回路図エディタで設計し,各部品はHDLで設計するのも良いでしょう
 - ✓ 全体の見通しがよくなります



- HDLからどんな回路が合成されるか知りたい
 - "Tools" -> "Netlist Viewers" -> "RTL Viewer" で 合成回路をなんとなく可視化できます
 - 自分のHDLがどのような回路に合成されるのか, HW設計とは なんたるかを知るのに有用な機能です
 - ✓ 今回の例題だとまったく面白くないですが,,,,

😣 🗐 🗊 RTL Viewer - /h	ome/lab3/takase/le3a/adder4bit/adder4bit - adder4bit	
<u>File Edit View T</u> ools	<u>W</u> indow <u>H</u> elp 🕏	Search altera.com
Page Title: adder4	Page: 1 of 1 💠 😰 🔩 🖶 🖶 🕄 😡 🥴	2 🖑 🔚 🚰 😩
Properties	₽@⊠ 🥰 adder4bit:1	
Selected block:		
		cout
		sum[30]
	ADDER	
	cin	



- シミュレーション
 - modelsim-aseの Transcript サブウィンドウに直接コマンドを 入力して実行できる. 慣れてきたらメニューをクリックするより 早い. コマンドヒストリも使える.
 - 実行履歴は "./simulation/modelsim/msim_transcript" に残るので,編集して別名で保存しておくとよい. Transcriptから do (ファイル名) で実行できる.
- モジュールについて
 - 回路全体ではなく一部だけを合成したりシミュレーションしたい 場合は、Project Navigatorの該当ファイルを右クリックして "Set as Top-Level Entity"を設定するとよい
- バージョン Quartus Prime 20.1 / ModelSim ASE 10.5b
 - Google先生に聞く時はツールのバージョンに注意すること. 古い/新しいために通用しないTIPSや解決策がある.

